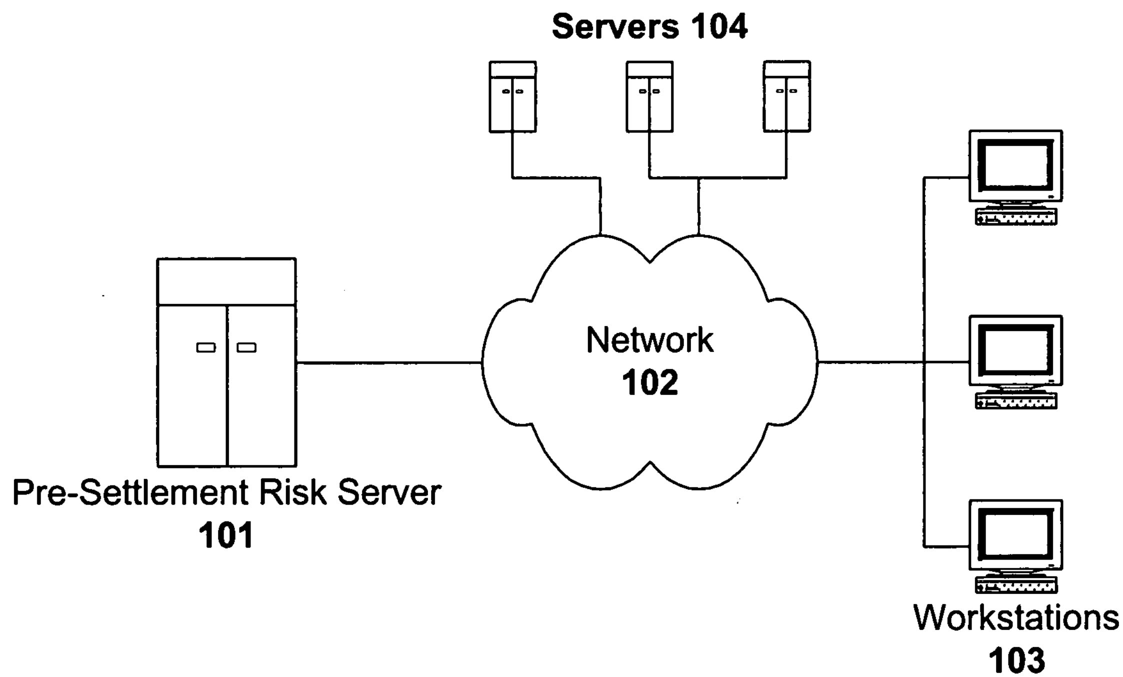

FIG. 1

$\Omega_d(X, Y)$

$N_{min} \leftarrow 2$

if $size[X] < N_{min}$ **or** $size[Y] < N_{min}$

  then return (0,0)

for each key $k \in X$

  do $D_x[X[k]] \leftarrow D_x[X[k]] + 1$

$MaxEntropy \leftarrow size[X] * log(size[D_x])$

for each key $k \in Y$

  do $D_y[Y[k]] \leftarrow D_y[Y[k]] + 1$

$Entropy \leftarrow 0$

for each key k of D_y

  do $Entropy \leftarrow Entropy - D_y[k] * log(D_y[k]/size[Y])$

return($MaxEntropy$, $Entropy$)

FIG. 2

```
Ωc(X, Y)
N_min ← 4
if size[X] < N_min
    then return (0,0)
n_x ← ⌈√size[X]⌉
X_sort ← sort[X]
for j ← 1 to n_x
    do Bounds[j – 1] ← X_sort[n_x * j – 1]
MaxEntropy ← n_x * log(√size[X])
MaxBounds ← Bounds[n_x – 1]
for each key v ∈ Y
    do for each key k of Bounds
        do if Y[v] < Bounds[k]
            then D_Y[k] ← D_Y[k] + 1; next
            if Y[v] > MaxBounds
                then D_Y[n_x – 1] ← D_Y[n_x – 1] + 1
                    next
for each key k of D_y
    do Entropy ← Entropy – D_y[k] * log(D_y[k]/size[Y])
return(MaxEntropy, Entropy)
```

FIG. 3

$\Omega_{c \times c}(X, Y)$

```
N_min ← 16
if size[X] < N_min
    then return (0,0)
n_x ← ⌈ ⁴√size[X] ⌉
X0_sort ← sort X by 0 dimension
Bounds0 ← FINDBOUNDARIES(X0_sort, dimension 0)
X1 ← COLLECTDATA(Bounds0, X0_sort)
X1_sort ← sort[X1]
Bounds1 ← FINDBOUNDARIES(X1_sort, dimension 1)
D_y ← BINPACK(Y, Bounds0, Bounds1)
MaxEntropy ← n_x * log( √size[X] )
for i ← 0 to n_x
    do for j ← 0 to n_x
        do Entropy ← Entropy–D_y[i][j]*log(D_y[i][j]/size[Y])
return(MaxEntropy, Entropy)
```

FIG. 4

```
Ω_cxd(X, Y)
N_min ← 2
if size[X] < N_min or size[Y] < N_min
    then return (0,0)
n_x ← size[X]
X0_sort ← sort X by 0 dimension
for each key k of X0_sort
    do XD[X0_sort[k][1]] ← XD[X0_sort[k][1]] + 1
n_m ← size[XD]
α ← log(n_m)/log(n_x)
n_c ← n_x^α
if n_c < N_min or n_m < N_min
    then return(0,0)
Bounds0 ← FINDBOUNDARIES(X0_sort, dimension 0)
MaxEntropy ← n_x * log(n_x^α * n_m)
D_y ← BINPACK(Y, Bounds0)
for i ← 0 to n_x
    do for each key j of D_y[i]
        do Entropy ← Entropy−D_y[i][j]*log(D_y[i][j]/size[Y])
return(MaxEntropy, Entropy)
```

FIG. 5

$\Omega_{d \times d}(X, Y)$

```
N_min ← 4
if size[X] < N_min or size[Y] < N_min
    then return (0,0)
for each key k ∈ X
    do D_x1[X[k][0]] ← D_x1[X[k][0]] + 1
       D_x2[X[k][1]] ← D_x2[X[k][1]] + 1
n_m ← size[D_x1] * size[D_x2]
MaxEntropy ← size[X] * log(n_m)
for each key k ∈ Y
    do D_y["Y[k][0]Y[k][1]"] ← D_y["Y[k][0]Y[k][1]"] + 1
Entropy ← 0
for each key k of D_y
    do Entropy ← Entropy – D_y[k] * log(D_y[k]/size[Y])
return(MaxEntropy, Entropy)
```

FIG. 6

Change	Semaphore	Odds favoring problem	Potential of problem (Maximum credible assessment)
Normal		3 to 1	Little potential of problem
Outer Normal		6 to 1	Substantial potential of problem
Borderline		20 to 1	Strong potential of problem
Abnormal		>20 to 1	Decisive potential of problem

FIG. 7

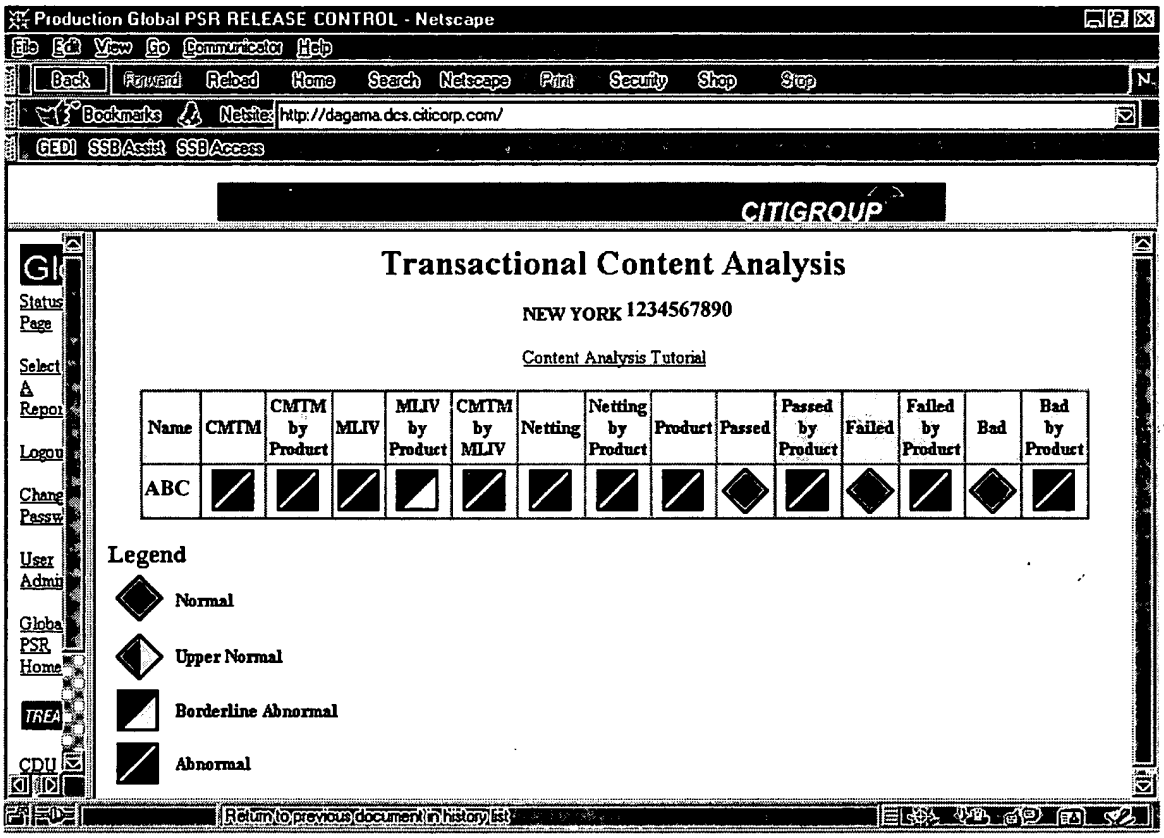
Production Global PSR RELEASE CONTROL - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://dagama.dcs.citicorp.com/
GEDI SSB Assist SSB Access
CITIGROUP
Status Page
Select A Report
Logout
Change Password
User Admin
Global PSR Home
TREA
CDU
Transactional Content Analysis
NEW YORK 1234567890
Content Analysis Tutorial
Name
CMTM
CMTM by Product
MLIV
MLIV by Product
CMTM by MLIV
Netting
Netting by Product
Product
Passed
Passed by Product
Failed
Failed by Product
Bad
Bad by Product
ABC
Legend
Normal
Upper Normal
Borderline Abnormal
Abnormal
Return to previous document in history list

FIG. 8

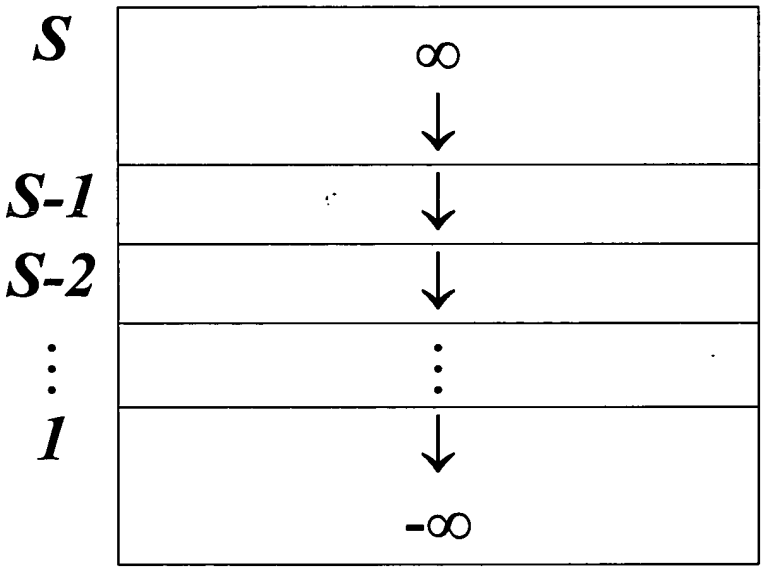

FIG. 9

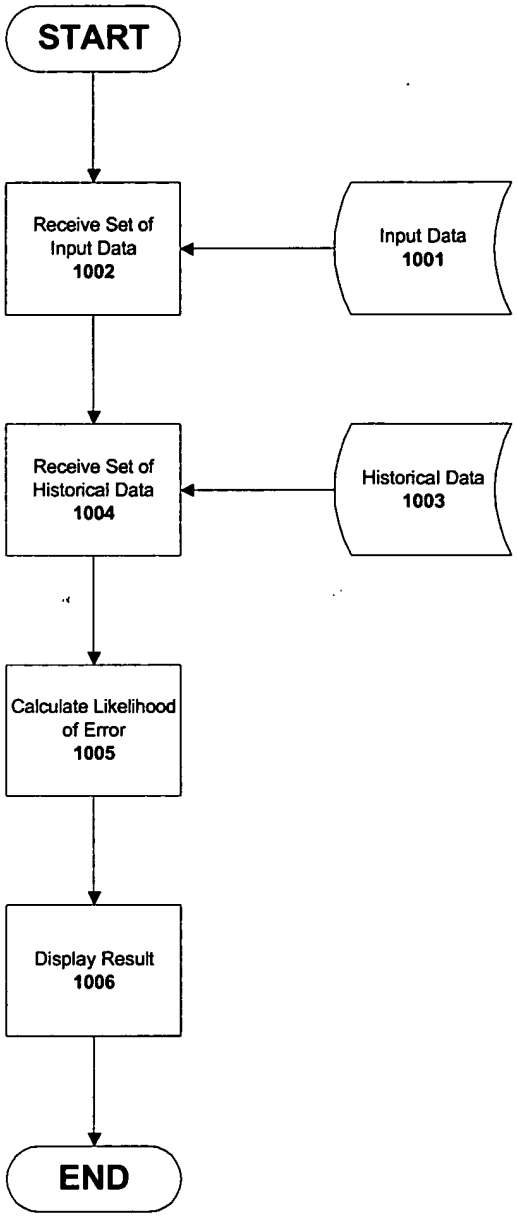
START
Receive Set of Input Data 1002
Input Data 1001
Receive Set of Historical Data 1004
Historical Data 1003
Calculate Likelihood of Error 1005
Display Result 1006
END

FIG. 10